



A-simple-Blockchain

19.05.2020

Names:

- 1) Ahmed Nasser Abdelkareem (9)
- 2) Belal Adel El Hadad (19)
- 3) Sarah Ehab Wahdan (29)
- 4) Abdelrahman Ahmed Mohamed Omran (36)
- 5) Abdelrahman Saeed (38)

Overview

It is required to implement a blockchain peer to peer network to handle a ledger of transactions collectively. A blockchain is a chain of blocks that continually grows as more transactions are added to the system. Each block references the block preceding it, and once the network agrees to a block, the transactions included in this block cannot be changed. The state transitions of this immutable ledger are preserved.

Goals

Each node in the network should:

1. holds a list of peers that contacted it lately such that from one node all the nodes in the network can be reached .
2. holds a consensus between the peers to agree on the collective set of transactions that reached the node during the previous round.
3. be able to verify the correctness of the transactions before voting on its inclusion in the ledger.
4. The ledger should not contain two conflicting transactions, i.e. if two conflicting transactions reach a node in a round only one should be included. A transaction conflicting with a past transaction should not be included in the ledger (this should be the vote of every honest node).
5. The required blockchain has relaxed requirements of the bitcoin network.

Specifications

1) Main Actors

- In BFT
 - 1) Client (Who parsing the transactions from the ledger and issues these transactions to each miner in the network.
 - 2) Miners
 - Primary Node
 - Secondary Node
- In Pow
 - 1) Client
 - 2) Miners

2) Code Structure

Our project consists of main components:

- 1) Network
 - A network layer that prepares the peer2peer connection between different nodes in the network.
 - Network starts a server for each node in a different thread.
 - Broadcasts messages to the other nodes in the network.
 - Listening to received blocks or transactions.
- 2) Node
 - The node is an abstraction of the machine in the network.
 - Each node reads the configuration file from the cloud
 - The configuration file contains specifications like:
 - Block size
 - Difficulty
 - Protocol (Pow,BFT)
 - List of IPs
 - Type of each node in the network (client or miner)
- 3) Transaction
 - Each transaction has
 - List of Inputs
 - List of outputs
 - Hash of transaction

4) BlockHeader

- Each header has
 - Nonce
 - Timestamp
 - Previous Block hash
 - Transactions hash (Merkle tree)
 - Hash of the block

5) Block

- Block Header
- List of transactions

6) Message

- Each message has:
 - 1) Type
 - Config message
 - NewBlock message
 - Preprepare message
 - Prepare message
 - Commit message
 - 2) Node publicKey
 - 3) Primary Node PublicKey
 - 4) View Number
 - 5) Sequence Number
 - 6) Block

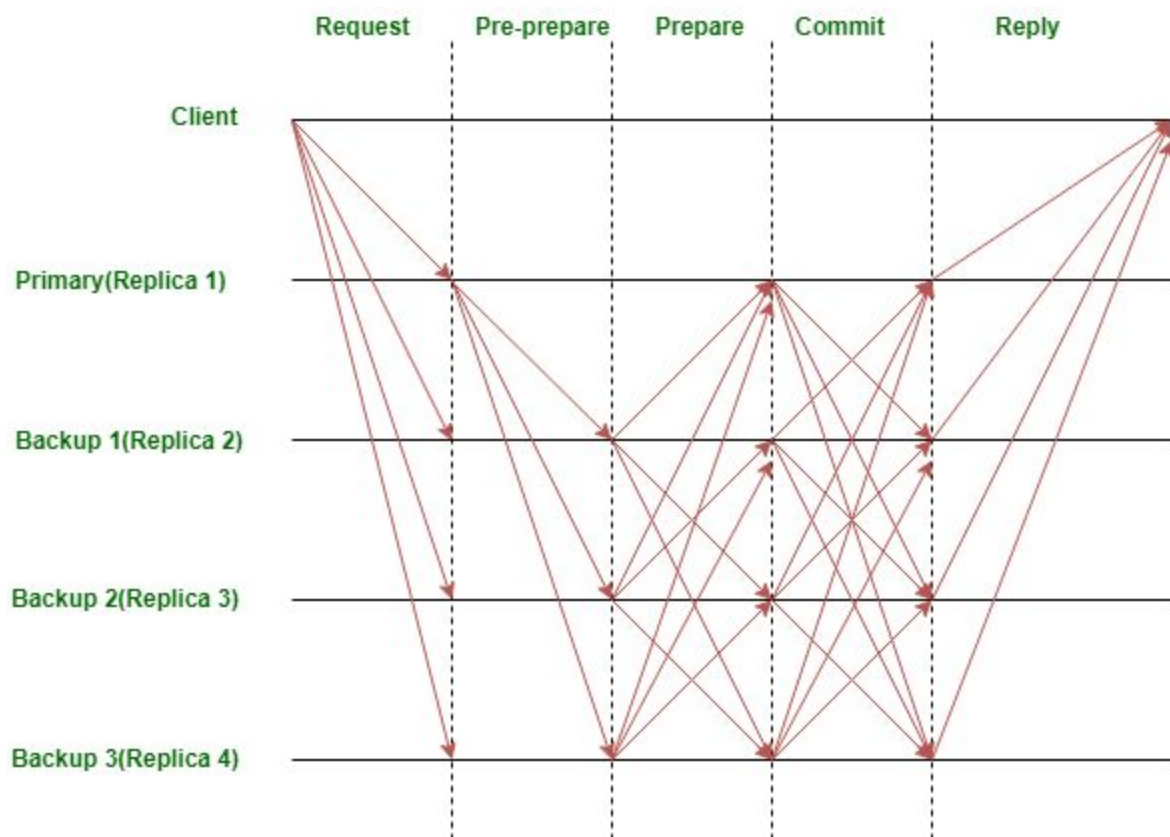
7) MessagePool

8) Analyser

- We use analyser to make a detailed report about our project like:
 - AvgMessageComplexity
 - NumberOfStaleBlocks
 - AvgTimeToMine
 - MessageComplexity
 - AvgTimeToAgreeOnBlock

3) How to run the code

- Each node (miner) should start
- The client (issuer) should start parsing line by line from the ledger
- The client starts to issue transactions to each node in the network.
- Each miner starts to receive transactions and verify them (Pow) till the Block size then create the block and generate a header for the block.
- After creating the block Primary Node starts to send the block to each secondary node.
- The secondary nodes starts to go through different phases



- Finally add the block to the chain.
- In Pow each miner receives the transactions and creates its block then starts mining till reaching the required difficulty.
- The first one -finishes mining- shares its block with other nodes.
- The other nodes received the block then added this block to its chain.

4) Assumption

- A transaction can depend on a transaction in the same block.

5) Role of each team member

Ahmed Nasser Abdelkareem was mainly responsible for the network and setting the configurations for the nodes.

Belal Adel El Hadad was responsible for implementing part of the Node and connecting that part with the network and that of BFT.

Sarah Ehab Wahdan was mainly responsible for implementing the logic of the BFT and mining.

Abdelrahman Ahmed Mohamed was responsible for implementing the Block and the Analyser.

Abdelrahman Saeed was responsible for the transaction issue and parts of the Node.

ScreenShots

Generating the keyPair for the current node :

```
node ip: 41.43.139.238
Node keys are generated
Node's public key: EC Public Key [bc:79:d1:2a:9a:36:59:0c:47:d6:01:5e:ba:4e:96:66:bf:e9:32:2f]
X: d8a8ef0516821775ddd7d179e107f3575a380e4b79acfc6a
Y: 7abe629a8fe6eb44d5e5707eb1b3cf72d127189eaddfd5eb

Node's private key: EC Private Key [bc:79:d1:2a:9a:36:59:0c:47:d6:01:5e:ba:4e:96:66:bf:e9:32:2f]
X: d8a8ef0516821775ddd7d179e107f3575a380e4b79acfc6a
Y: 7abe629a8fe6eb44d5e5707eb1b3cf72d127189eaddfd5eb

entered issuer1
```

Pow :Hashing the block till the right difficulty (here difficulty =1):

```
212
id 58
transaction accepted
block hashed ..
Working in pow
block hashed ..
block hashed ..
block hashed ..
block hashed ..
block hashed ..
block is mined...
block hash is: 098a205bfc3107aed89c575ff73bc51f1676462d12f155c59bd711e70fd819ec
chain size: 4
```

Bft :

Generating the keyPair for the current node :

```
node ip: 41.43.139.238
Node keys are generated
Node's public key: EC Public Key [89:16:d6:40:6a:34:5a:56:0e:c0:7c:eb:e3:72:5e:6f:3f:b2:d3:ae]
X: bd3d9d8d13ef084392526d6b8c40e4cb90143177c964622f
Y: 1d0b40d02f696620db2f3bc28c798aa8dbb617aa83e37c56

Node's private key: EC Private Key [89:16:d6:40:6a:34:5a:56:0e:c0:7c:eb:e3:72:5e:6f:3f:b2:d3:ae]
X: bd3d9d8d13ef084392526d6b8c40e4cb90143177c964622f
Y: 1d0b40d02f696620db2f3bc28c798aa8dbb617aa83e37c56

entered issuel
is primary constructor :true
```

The phases of BFT:

```

transcation accepted
block hashed ..
PBFT creating block transactions
new Block hash: 94107e4b1f42bb67cf23dlf24fd919145553a73ff9a08b6ee9d53268630043dd
new block this.seqNum 1
primary public key: EC Public Key [24:fb:90:cf:b0:70:9c:bd:f5:9a:98:7e:04:58:ce:90:f4:31:3b:f1]
    X: 7c9122d8a4ee0d02e160639909ad960db68898b92cc917d2
    Y: 595d3754284c89dcf58950acb16913f7203f0e0a540fe3b1

new block is created
new Block node public key: EC Public Key [24:fb:90:cf:b0:70:9c:bd:f5:9a:98:7e:04:58:ce:90:f4:31:3b:f1]
    X: 7c9122d8a4ee0d02e160639909ad960db68898b92cc917d2
    Y: 595d3754284c89dcf58950acb16913f7203f0e0a540fe3b1

node signature is generated.
Node's signature: [B@404f273f
pre-prepare message is created
node signature in generate pre-prepare:[B@404f273f
primary node public key: EC Public Key [24:fb:90:cf:b0:70:9c:bd:f5:9a:98:7e:04:58:ce:90:f4:31:3b:f1]
    X: 7c9122d8a4ee0d02e160639909ad960db68898b92cc917d2
    Y: 595d3754284c89dcf58950acb16913f7203f0e0a540fe3b1

primary node public key: EC Public Key [24:fb:90:cf:b0:70:9c:bd:f5:9a:98:7e:04:58:ce:90:f4:31:3b:f1]
    X: 7c9122d8a4ee0d02e160639909ad960db68898b92cc917d2
    Y: 595d3754284c89dcf58950acb16913f7203f0e0a540fe3b1

node signature in generate pre-prepare: [B@404f273f

```

```

id 81
transaction rejected
type : new block
New block is received with seq num : 1
passing set new block validation
116
id 91
transcation accepted
type : pre-prepare
preprepare message max malicious nodes: 0
preprepare message block hash: f23b194707849140a7204a0a5c3e7385e14232d9a7ffa45d6222465170d6ae53
preprepare message type: pre-prepare
preprepare message sending node public key: EC Public Key [d9:77:5e:21:d1:c0:75:b0:f3:9b:39:b0:bc:62:69:f7:5b:a4:cb:21]
    X: ecda01b0e9e22fa0064b32d0b98182454379b947328f40f4
    Y: 87754c9609bf806e81bc9d486f72a15f47a93c3b249a698f

preprepare message primary public key the same the above: EC Public Key [d9:77:5e:21:d1:c0:75:b0:f3:9b:39:b0:bc:62:69:f7:5b:a4:cb:21]
    X: ecda01b0e9e22fa0064b32d0b98182454379b947328f40f4
    Y: 87754c9609bf806e81bc9d486f72a15f47a93c3b249a698f

preprepare message seq num: 1
preprepare message view num: 2
verify peer signature : true
primary ley for the current node: EC Public Key [d9:77:5e:21:d1:c0:75:b0:f3:9b:39:b0:bc:62:69:f7:5b:a4:cb:21]
    X: ecda01b0e9e22fa0064b32d0b98182454379b947328f40f4
    Y: 87754c9609bf806e81bc9d486f72a15f47a93c3b249a698f

node view num: 2

```

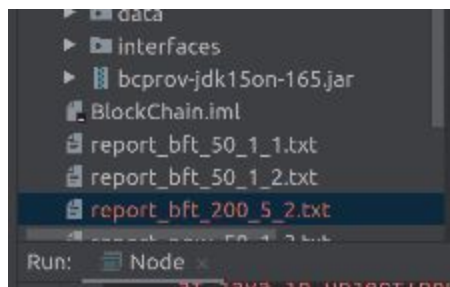


```
id 141
transaction rejected
219
id 63
transcation accepted
type : prepare
Entered prep
IN PRE2 2
prepare validation is passed
id 91
node passed prepare phase
node generated commit message and added it to her commit pool
```

```
type : commit
commitMessage.getMessageType() commit
this.state prepare
commitMessage.getViewNum() 2
this.viewNum 2
commitMessage.verifyPeerSignature() true
!commitPool.isMessageExists(commitMessage) true
commit validation is passed
id 49
node passed commit phase
chain size: 1
node added the block to chain1
```

Analysis results :

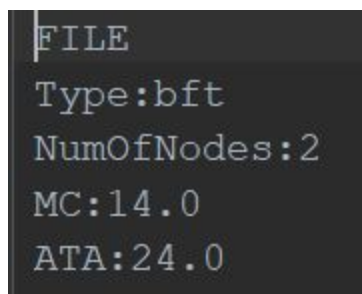
We calculate the analytics for each node during processing then when we finish each node broadcasts its analytics to the other nodes and receives theirs , then they calculate the overall analytics and then it prints a file with the required calculations.



For bft with 2 nodes

message complexity = 14 message;

Average time to agree on a block = 24 ms;




And with more blocks :

For bft with 2 nodes :

message complexity = 9.5 message;

Average time to agree on a block = 29.3333 ms;



```
FILE
Type:bft
NumOfNodes:2
MC:9.5
ATA:29.333333333333332
|
```

And with 3 nodes and more issued transactions (more created blocks):

```
FILE
Type:bft
NumOfNodes:3
MC:9.2
ATA:4.5
```

For pow with size 50 , difficulty =1

Avg message complexity = 0 message(it gave 0.5 later);

Number of stale blocks = 3 blocks;

Average time to mine = 2 ms;

```
FILE
Type:pow
BlockSize:50
Difficulty:1
AMC:0.0
NSB:3.0
ATM:2.0
```

When we change the difficulty to 5 the average time to mine increases to 5 sec.

And the number of stalls is 0 because we issued less transactions. And created around 2 blocks.

```
BlockSize:200
Difficulty:5
AMC:0.25
NSB:0.0
ATM:5043.0
```

With a malicious node :

Malicious in pbft means that the node will say no in the voting, we implement it by making it refrain from broadcasting here commit message to other nodes in the network so If the total number of commit messages in each node's pool is greater than or equal $2 * f + 1$ (f : max number of malicious nodes the pbft can handle) it will add the block to its chain, on the other hand If the total number of commit messages in each node's pool is less $2 * f + 1$, the node will ignore the block.

With one node as malicious the block was accepted and added to the chain:

```
node passed commit phase  
chain size: 1  
Analytics sent ...  
node added the block to chain1
```

With one node as malicious in 2 node network the block won't added to chain:

```
node passed prepare phase  
node generated commit message and added it to her commit pool
```